

**IN THE UNITED STATES PATENT AND TRADEMARK OFFICE**  
**BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES**

In re application of:  Sunil Kunisetty, et al.  Serial No.: 10/051,274  Filed on: January 22, 2002	Confirmation No.: 7586  Examiner: Chrystine Pham  Group Art Unit No: 7586
--	---

For: METHOD AND SOFTWARE FOR PROCESSING SERVER PAGES

**Mail Stop Appeal Brief – Patents**

Commissioner for Patents

P.O. Box 1450

Alexandria, VA 22313-1450

**REPLY BRIEF**

Sir:

This Reply Brief is filed in reply to the Examiner's Answer mailed on November 14, 2007.

**I. STATUS OF CLAIMS**

The status of the claims has not changed since the filing of the Appeal Brief on June 18, 2007 and remains as:

Claims 1-2, 5-6, 9, 11, 15-17, 19-21, and 23-34 are pending in this application, were finally rejected in the Final Office Action mailed on October 20, 2006 (hereinafter "the Final Office Action") and are the subject of this appeal.

## II. GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

The grounds of rejection to be reviewed on appeal have not changed since the filing of the Appeal Brief on June 18, 2007 and remain as:

Claims 1-2, 5-6, 9, 11, 15-17, 19-21, and 23-34 stand rejected under 35 U.S.C. § 103(a) as being allegedly unpatentable over *Agrawal et al.*, U.S. Patent Application 2002/0004813 A1 (hereinafter “*Agrawal*”) in view of *Claussen et al.*, U.S. Patent No. 6, 675, 354 B1 (hereinafter “*Claussen*”).

## III. ARGUMENTS

### Introduction

It is respectfully submitted that a sufficient factual basis has not been proffered during the prosecution of the present application to support the rejection of Claims 1-2, 5-6, 9, 11, 15-17, 19-21, and 23-34 as being unpatentable over *Agrawal* in view of *Claussen*.

This is in response to the Examiner’s Answer mailed on November 14, 2007

### Applicants’ arguments are not “piecemeal”.

Among other features, Claim 1 recites “instantiating a distinct instance of the servlet class on the server, wherein instantiating each instance of the servlet class does not create another copy of the markup text.”

In the Appeal Brief, the Applicants explained that “[n]either *Agrawal* nor *Claussen* discloses, teaches, or suggests anything about “instantiating” a servlet class without creating “another copy of the markup text,” and that “[t]his notion is not found anywhere in the cited references.” (Appeal Brief, page 10, paragraph 1.)

The Examiner alleges that this argument is based on a “piecemeal analysis,” (i.e. the analysis based on attacking each of the references individually). This is wrong.

The most common, efficient and clear-cut way to show that a combination of prior art references does not satisfy a claim limitation is to show that none of the references satisfies the claim limitation. The logic is simple - if SET A does not have element X, and SET B does not have element X, then the union of SET A and SET B cannot possibly have element X. Thus, a

combination of the references does not satisfy the particular claim limitation when the references are both individually unable to satisfy the particular limitation.

Hence, once the Applicants show that neither *Agrawal* nor *Claussen* discloses, teaches, nor suggests anything about the “instantiating” a distinct instance of the servlet class without creating “another copy of the markup text,” as disclosed in Claim 1, a combination of *Agrawal* and *Claussen* is incapable of satisfying that particular limitation of Claim 1.

Therefore, Applicants have chosen to show that neither *Agrawal* nor *Claussen* discloses, teaches, nor suggests the aforementioned limitation of Claim 1. As it will be shown below, neither *Agrawal* nor *Claussen* discloses, teaches, nor suggests the aforementioned limitation of Claim 1.

**The Combined Teaching Of Agrawal And Claussen Does Not Teach “instantiating a distance instance of the servlet class on the server” Without Creating Another Copy Of The Markup Text, As Recited In Claim 1.**

As was stated above, Applicants will proceed with this argument by showing the following steps: because i) *Agrawal* does disclose the above limitation, and ii) *Claussen* does not disclose the above limitation, hence, iii) the combination of *Agrawal* and *Claussen* does not teach the above limitation.

i) *Agrawal* does not disclose “instantiating” a servlet class without creating “another copy of the markup text.”

In her Answer, the Examiner concedes that although *Agrawal* discloses dynamically generating HTML pages without generating (i.e. creating) another copy of the static HTML (i.e. static markup text) which has been retrieved from the cache memory, *Agrawal* does not expressly disclose said dynamically generated (i.e. instantiated) document as an instance of a “servlet class” *per se*. (Examiner’s Answer, page 10, lines 10-13.) Applicants wish to emphasize the importance of that assertion, and focus on the contention that *Agrawal*’s dynamically generating HTML pages does not disclose Applicants’ “instantiating” a servlet class.

The way *Agrawal* dynamically generates HTML pages is that upon a client's request, *Agrawal* first retrieves blocks of the code from a cache (wherein those blocks are used to generate the text and/or graphics as it appears on the web page, and wherein those blocks are often repeated across multiple pages in an unchanged form). Then, *Agrawal* generates a compiled page based on the retrieved blocks. (*Agrawal*, Summary, paragraphs [0013]-[0015].) *Agrawal* retrieves those blocks without "instantiating" any of the servlet classes. (*Agrawal*, Summary, paragraphs [0013]-[0015].)

In contrast, when Applicants generate HTML pages upon a client's request, Applicants do the following: 1) first, Applicants "instantiate" a distinct instance of the servlet class on the server, 2) then, Applicants execute said distinct instance of the servlet class. The said execution involves 1) generating a compiled page based on the copy of the markup text that resides in shared memory, and 2) sending the compiled page [including a "markup text"] to a client that requested the page. (Claim 1)

"Instantiating" of the distinct instance of the servlet class on the server is the process of creating a specific object which is an instance (i.e. a copy) of a class. "Instantiating" involves creating a physical instance of a declared object or class. In another words, creating an "instance" requires making a copy of another piece of code (let call it a "parent code"), so that the copy (i.e. the instance) of the parent code is capable of executing the same commands as those included in the parent code.

Here, Applicants create an "instance" of a code which upon a client's request will retrieve the "markup text" from the shared memory. Hence, Applicants' instance retrieves a particular "markup text" from the shared memory.

Such an instance is not disclosed in *Agrawal*. *Agrawal* does not disclose creating any of the instances of a servlet class, and does not disclose creating any of the objects to handle blocks of code to be delivered to a client upon a client's request. Since *Agrawal* does not teach about "instantiating" servlet class, *Agrawal* cannot teach about "instantiating" a servlet class without creating "another copy of the markup text." Hence, *Agrawal* does not disclose Claim 1 limitation about "instantiating" a servlet class without creating "another copy of the markup text."

ii) Claussen does not disclose “instantiating” a servlet class without creating “another copy of the markup text”

In her Answer, the Examiner alleges that because *Claussen* 1) discloses techniques for dynamically serving web page content (i.e., “**dynamically generated HTML**”); 2) discloses the JSP (“Java server Page) as **web templates** that enable Java code to be embedded in **static HTML** to be served in response to a client browser request; and 3) discloses translating the flat web page into the servlet, compiling the servlet to generate the servlet class, loading the generated class and invoking the servlet “to cause given web content to be returned to the requesting browser;” hence it is clear that the “customized web content” teaches “an instance” (that is different from other customized instances) of the generated servlet class. (Examiner’s Answer, pages 10-11.)

Moreover, the Examiner alleges that because *Claussen* discloses loading class for the generated servlet at page translation time, hence *Claussen* discloses dynamically generating HTML pages in the context of dynamically generating an instance (i.e. instantiating) of the servlet class. (Examiner’s Answer, page 11.) This is incorrect.

What *Claussen* describes is only a typical mechanism used to generate typical servlet classes. (*Claussen*, Fig.2, column 5, lines 1-45.) Just because the *Claussen* reference describes a typical mechanism used to generate typical servlet classes, it does not mean that *Claussen* discloses Applicants’ unique instances of servlet classes that are capable of creating “another copy of the markup text.” A typical servlet class and a typical instance of the servlet classes, both described in *Claussen*, are different from Applicants’ servlet class and Applicants’ instance because a typical class and instance are not created to handle a “markup text” without making “another copy of the markup text.” (Claim 1) Therefore, *Claussen* does not teach about “instantiating” servlet class having the limitations disclosed in Claim 1.

iii) Combination of *Agrawal* and *Claussen* does not disclose “instantiating” a servlet class without creating “another copy of the markup text”

As stated above, the most efficient way to show that a combination of prior art references does not satisfy the claim limitations is to show that none of the references satisfies a particular claim limitation or particular limitations.

Based on the above discussion, Applicants maintain that they have shown that neither *Agrawal* nor *Claussen* discloses, teaches, nor suggests anything about the “instantiating” a distinct instance of the servlet class without creating “another copy of the markup text,” as disclosed in Claim 1. Therefore, a combination of *Agrawal* and *Claussen* is incapable of satisfying that particular limitation of Claim 1.

Therefore, Applicants respectfully request that the rejection of Claim 1 under 35 U.S.C. § 103(a) be withdrawn.

**“Static elements within a servlet class”**

Applicants concede that they did not recite the following phrase in their Claim1: “static elements within a servlet class.” However, the issue of retrieving cached blocks of the web page, whether the blocks contain static elements within a servlet or not, became a side issue now since Applicants believe that they have already shown that *Agrawal* does not disclose “instantiating a distinct instance of the servlet class on the server” without creating another “copy of the markup text.” Regardless of whether the blocks contain static elements within a servlet class or not, the Applicants have already shown that *Agrawal* does not disclose all the features which are disclosed in Claim1. As was shown above, *Agrawal* does not disclose “instantiating” a servlet class without creating “another copy of the markup text” regardless of whether the markup text contains static elements within a servlet class because *Agrawal* just does not create the instances which are disclosed by Applicants in Claim 1.

Therefore, Applicants believe that they have already provided above a sufficient basis for the patentability of Applicants’ Claim 1.

**(Dependent Claims 23-25) - Claussen Does Not Disclose “servlet class includes inner class” Disclosed In Claim 23.**

The Examiner alleges that because *Claussen* discloses inserting methodDefinition element as a child (inner element) of the root element (outer element) of the HTML document, and discloses embedding one scripting language (i.e. inner element) within another scripting language (i.e. outer element), *Claussen* clearly teaches the “inner class.” (Examiner’s Answer, page 12, paragraph 3.) Applicants respectfully disagree.

The Applicants would like to reiterate and point out that the cited portions of *Claussen* only refer to techniques for using multiple scripting languages to generate pages. The cited portions of *Claussen* do not describe generating an inner class for the servlet class as claimed in Applicants’ Claim 23 (which depends from Claim 1) because Applicants’ servlet class, as claimed in the independent Claim 1, is created by “instantiating a distance instance of the servlet class on the server” without creating “another copy of the markup text.” Because *Claussen* does not disclose instances of the servlet class on the server instantiated without creating another copy of the markup text, *Claussen* cannot disclose “inner class” embedded within the instance of the servlet class which is “instantiated” without creating another copy of the markup text. Hence, *Claussen* does not disclose “servlet class includes inner class” disclosed in Claim 23.

**Because None Of The Cited Portions From Agrawal Actually Teach An Inner Class, And None Of The Cited Portions From Claussen Teach An Inner Class, A Combined Teaching Of Agrawal And Claussen Does Not Teach An “inner class.”**

In her Answer, the Examiner alleged that even though *Agrawal* does not teach the “inner class,” *Agrawal* discloses a shopping cart (i.e. a class element) containing item descriptions and/or user comments on the items, which are cached and subsequently retrieved upon request from multiple users, and *Claussen* teaches **embedding one element** of one scripting language **into another element** of another scripting language in the same servlet, it would be obvious that the shopping cart class (defined in one scripting language) containing the cached static item descriptions (which are the same for all users) can be embedded in the servlet class (defined in another scripting language), and the shopping cart class can be instantiated for each different user requesting an instance of the shopping cart. (Examiner’s Answer, page 13, paragraph 1.) Applicants respectfully disagree.

Again, the best way to show that a combination of references does not satisfy the particular claim limitations is to show that none of the references satisfies the particular claim limitations. In another words, a combination of the references would not satisfy the particular claim limitations when the references are both individually unable to satisfy the particular claim limitations.

Therefore, Applicants will proceed with this argument by showing the following steps: because i) *Agrawal* does disclose the above limitation, and ii) *Claussen* does not disclose the above limitation, hence, iii) the combination of *Agrawal* and *Claussen* does not teach the above limitation.

None of the cited portions from *Agrawal* actually teaches an inner class

The Examiner has conceded that *Agrawal* was not relied upon to teach the “inner class.” . (Examiner’s Answer, page 12, last paragraph.) Further, the Examiner asserted that *Agrawal* only discloses a shopping card containing item descriptions and/or user comments on the items. (Examiner’s Answer, page 13, paragraph 1.) However, *Agrawal* does not actually teach an inner class because the item description and/or user comments on the items are not actually an array of characters which can be accessed by separate instances of an application. *Agrawal*’s item description and/or comments are accessible only to one code, i.e. the code that generates the Web Page. Thus, *Agrawal* does not actually teach Applicant’s inner class.

Combined teaching of *Agrawal* and *Claussen* does not teach an “inner class”

As it was shown above, Applicants maintain that they have shown that neither *Agrawal* nor *Claussen* discloses, teaches, nor suggests anything about the “inner class” disclosed in Claim 23. Therefore, a combination of *Agrawal* and *Claussen* is incapable of satisfying that particular limitation of Claim 23.



**(Independent Claim 5) The Combined Teaching Of *Agrawal* And *Claussen* Does Not Teach “instantiating a distance instance of the servlet class on the server” without creating another copy of the markup text,” as recited in Claim 5.**

The Examiner asserted that the combined teaching of *Agrawal* and *Claussen* teaches “accessing the set of markup test in a shared, read-only memory when the code from the first instance of the applicant is executed.” The Applicants do not challenge this assertion. However, the combined teaching of *Agrawal* and *Claussen* does not teach “instantiating a distance instance of the servlet class on the server” without creating another “copy of the markup text,” as recited in Claim 5.

Therefore, Applicants maintain that they have shown that neither *Agrawal* nor *Claussen* discloses, teaches, nor suggests anything about the “instantiating” a distinct instance of the servlet class without creating “another copy of the markup text,” as disclosed in Claim 5. Thus, a combination of *Agrawal* and *Claussen* is incapable of satisfying that particular limitation of Claim 5.

Therefore, Applicants respectfully request that the rejection of Claim 5 under 35 U.S.C. § 103(a) be withdrawn.

#### IV. CONCLUSION AND PRAYER FOR RELIEF

Based on the foregoing, it is respectfully submitted that the rejections of Claims 1, 2, 5-6, 9, 11, 15-17, 19-21, and 23-34 lack the requisite factual and legal bases. Appellants respectfully request that the Honorable Board **reverse** the rejections of Claims 1, 2, 5, 6, 9, 11, 15-17, 19-21, and 23-34.

Respectfully submitted,

HICKMAN PALERMO TRUONG & BECKER LLP

Date: January 14, 2008

/BrianDHickman35894/  
Brian D. Hickman  
Registration No. 35,894

2055 Gateway Place, Suite 550  
San Jose, California 95110-1089  
Tel: (408) 414-1224  
Fax: (408) 414-1076